

PROGRAM CODE

```
#include<stdio.h>
```

```
int pn;  
int pno[10],at[10],bt[10],ct[10],tat[10],wt[10],rt[10];  
float avg_waiting,avg_turnaround,avg_response;  
int twt = 0,ttat = 0,trt = 0,curt = 0;
```

```
void sortAT();  
void sortBT();  
void calc_SJF();  
void printTable();
```

```
void main()  
{  
    printf("Number of Processes = ");  
    scanf("%d", &pn);  
  
    for (int i = 0; i < pn; i++)  
    {  
        pno[i]=i+1;  
        printf("[Processes %d][AT] - ",i+1);  
        scanf("%d", &at[i]);  
        printf("[Processes %d][BT] - ",i+1);  
        scanf("%d", &bt[i]);  
    }  
    sortAT(pn);  
    sortBT(pn);  
    calc_SJF(pn);  
    printTable(pn);  
}
```

```
void sortAT(int n)  
{  
    for (int i = 0; i < n - 1; i++)  
    {  
        for (int j = 0; j < n - i - 1; j++)  
        {  
            if (at[j] > at[j + 1])  
            {  
                int temp_pno = pno[j];  
                pno[j] = pno[j + 1];  
                pno[j + 1] = temp_pno;  
  
                int temp_at = at[j];  
                at[j] = at[j + 1];  
                at[j + 1] = temp_at;  
  
                int temp_bt = bt[j];  
                bt[j] = bt[j + 1];  
                bt[j + 1] = temp_bt;  
            }  
        }  
    }  
}
```

```

void sortBT(int n)
{
    for (int i = 1; i < n - 1; i++)
    {
        for (int j = 1; j < n - i; j++)
        {
            if (bt[j] > bt[j + 1])
            {
                int temp_pno = pno[j];
                pno[j] = pno[j + 1];
                pno[j + 1] = temp_pno;

                int temp_at = at[j];
                at[j] = at[j + 1];
                at[j + 1] = temp_at;

                int temp_bt = bt[j];
                bt[j] = bt[j + 1];
                bt[j + 1] = temp_bt;
            }
        }
    }
}

void calc_SJF(int n)
{
    int completed=0;
    for (int i = 0; i < n; i++) {
        ct[i] = curt + bt[i];
        tat[i] = ct[i] - at[i];
        wt[i] = tat[i] - bt[i];
        rt[i] = wt[i];
        curt += bt[i];
        completed++;

        twt += wt[i];
        ttat += tat[i];
        trt += rt[i];
    }
    avg_waiting = (float)twt / n;
    avg_turnaround = (float)ttat / n;
    avg_response = (float)trt / n;
}

void printTable(int n)
{
    printf("\n+-----+-----+-----+-----+-----+\n");
    printf("| Process ID | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time | Response Time |\n");
    printf("+-----+-----+-----+-----+-----+-----+\n");
    for (int i = 0; i < n; i++)
    {

```

```

    printf("| %10d | %12d | %11d | %15d | %14d | %13d | %14d |\n", pno[i], at[i], bt[i], ct[i], tat[i], wt[i], rt[i]);
}
printf("+-----+-----+-----+-----+-----+-----+-----+\n");
printf("\nAverage Turn Around Time = %.2f\n", avg_turnaround);
printf("Average Waiting Time = %.2f\n", avg_waiting);
printf("Average Response Time = %.2f\n", avg_response);
}

```

OUTPUT

```

ubuntu@administrator-hcl-desktop: ~/Desktop/gopikrishna
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$ gcc sjf_at.c
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$ ./a.out
Number of Processes = 5
[Processes 1][AT] - 3
[Processes 1][BT] - 3
[Processes 2][AT] - 1
[Processes 2][BT] - 2
[Processes 3][AT] - 5
[Processes 3][BT] - 1
[Processes 4][AT] - 0
[Processes 4][BT] - 5
[Processes 5][AT] - 2
[Processes 5][BT] - 3

```

Process ID	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
4	0	5	5	5	0	0
3	5	1	6	1	0	0
2	1	2	8	7	5	5
5	2	3	11	9	6	6
1	3	3	14	11	8	8

```

Average Turn Around Time = 6.60
Average Waiting Time = 3.80
Average Response Time = 3.80
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$

```